

## ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 53.083.9

DOI: 10.17223/19988605/52/15

**В.М. Соловьев, Д.В. Сперанский**

### ОЦЕНКА КАЧЕСТВА ДИАГНОСТИЧЕСКИХ ТЕСТОВ НА МОДЕЛЯХ ЦИФРОВЫХ УСТРОЙСТВ ISCAS

Рассматривается методика оценки качества диагностических тестов на моделях цифровых устройств, стандартизованных международным симпозиумом IEEE. Методика ориентирована на использование современных программных средств автоматизированного проектирования электронных устройств.

**Ключевые слова:** цифровые устройства (ЦУ); языки описания ЦУ Verilog, VHDL; контроль и диагностика ЦУ; программная среда автоматизированного проектирования ЦУ.

В настоящее время интенсивно идет процесс внедрения во все области человеческой деятельности электронных изделий, которые являются основой цифровизации общества. Эти электронные изделия (цифровые устройства, ЦУ) нужно проектировать, изготавливать и эксплуатировать, поддерживая в рабочем состоянии. На каждом этапе жизненного цикла ЦУ требуется знать их техническое состояние, что обеспечивается контролем и диагностированием, подачей воздействий (теста) и анализом реакции изделия на него. Создание эффективных тестов является одной из актуальных задач современной диагностики. Эффективность тестов зависит от используемых алгоритмов их создания, поэтому это направление технической диагностики активно развивается. Появляется большое количество таких алгоритмов, продукция которых (тесты) требует анализа и оценки качества. Для объективного анализа и оценки алгоритмов экспертами разработаны тестовые схемы ЦУ, стандартизованные международным симпозиумом IEEE по схемам и системам ISCAS [1]. Используя эти тестовые схемы, можно в среде автоматизированного проектирования электронных устройств (Electronic Design Automation, EDA) создавать алгоритмические модели ЦУ ISCAS [1], почти не отличающиеся от их реальных физических моделей.

Современные EDA, как правило, позволяют создавать алгоритмические модели на языках описания электронных схем Verilog или VHDL. Язык Verilog ближе к ЦУ, быстрее и современнее VHDL. В то же время ранее созданные модели на VHDL могут транслироваться в Verilog с последующей компиляцией, что и обеспечивает их совместимость. При этом появляется возможность эффективно исследовать различные подходы и идеи, лежащие в основе алгоритмов синтеза тестов для сложных ЦУ. Кроме того, такой подход позволяет оценить техническое состояние ЦУ, создать удобный аппарат для получения, хранения и анализа результатов диагностических экспериментов и, что особенно важно, представить эти результаты (scientific data) в различной удобной для исследователя форме (битовым массивом, бинарным вектором, графическими изображениями и т. д.). Все это делает EDA базой для разработки новой перспективной методики, решающей широкий круг задач технического диагностирования сложных ЦУ. Демонстрировать возможности такой методики можно на стандартных тестовых схемах ISCAS. Для них предлагаемая методика может содержать следующие этапы: выбор тестовой схемы из состава ISCAS; ввод исходных данных (принципиальная схема или описание ЦУ на языках Verilog, VHDL) и параметров в среде моделирования (например, Quartus II [2],

ModelSim [3], Icarus Verilog (Iverilog) [4] и т.д.); разработка на основе анализируемых алгоритмов контрольных и диагностических тестов (testbench) в среде моделирования; компиляция и отладка созданной модели ЦУ и testbench; диагностические эксперименты (simulation) с отлаженной моделью ЦУ и тестами в среде моделирования, а также визуализация результатов диагностического эксперимента; анализ полученных результатов и размещение результатов в открытом доступе (например, GitHub) для обеспечения возможности их проверки и повторения [5].

## 1. Подготовка и проведение диагностических экспериментов

Использование тестовых цифровых устройств, рекомендованных IEEE (ISCAS-85, ISCAS-89, ISCAS-99) [1], позволяет создавать одинаковые транспарентные условия для проведения диагностических экспериментов. Это обеспечивает объективную оценку эффективности контрольных и диагностических тестов, создаваемых на основе предлагаемых алгоритмов. Кроме того, такой подход решает и целый ряд других задач диагностирования, например облегчает (формализует) построение входных тестовых последовательностей в среде автоматизированного проектирования электронных устройств (EDA) и их отладку. К сожалению, автоматизированная генерация диагностической среды (testbench) в EDA до сих пор не решена. Разработчики ЦУ вынуждены создавать ее в «ручном» режиме, используя для построения, например, языки программирования высокого уровня и вводить затем тесты в EDA практически вручную. Для небольших ЦУ, содержащих до 100 элементов (вентилей, комбинационных схем) это еще может быть как-то оправдано. Однако при большом количестве входов-выходов и элементов в ЦУ (а именно такие тестовые схемы и входят в ISCAS) создание и ввод диагностических последовательностей, ориентированных даже на константные неисправности, является трудоемкой задачей. Поэтому чаще всего исследователи используют свои «закрытые» методики, представляя в публикациях результаты моделирования таблицами, проверить которые не представляется возможным.

Моделирование в среде EDA – это практически создание алгоритмической модели с высокой степенью адекватности реальному (физическому) ЦУ (рис. 1).

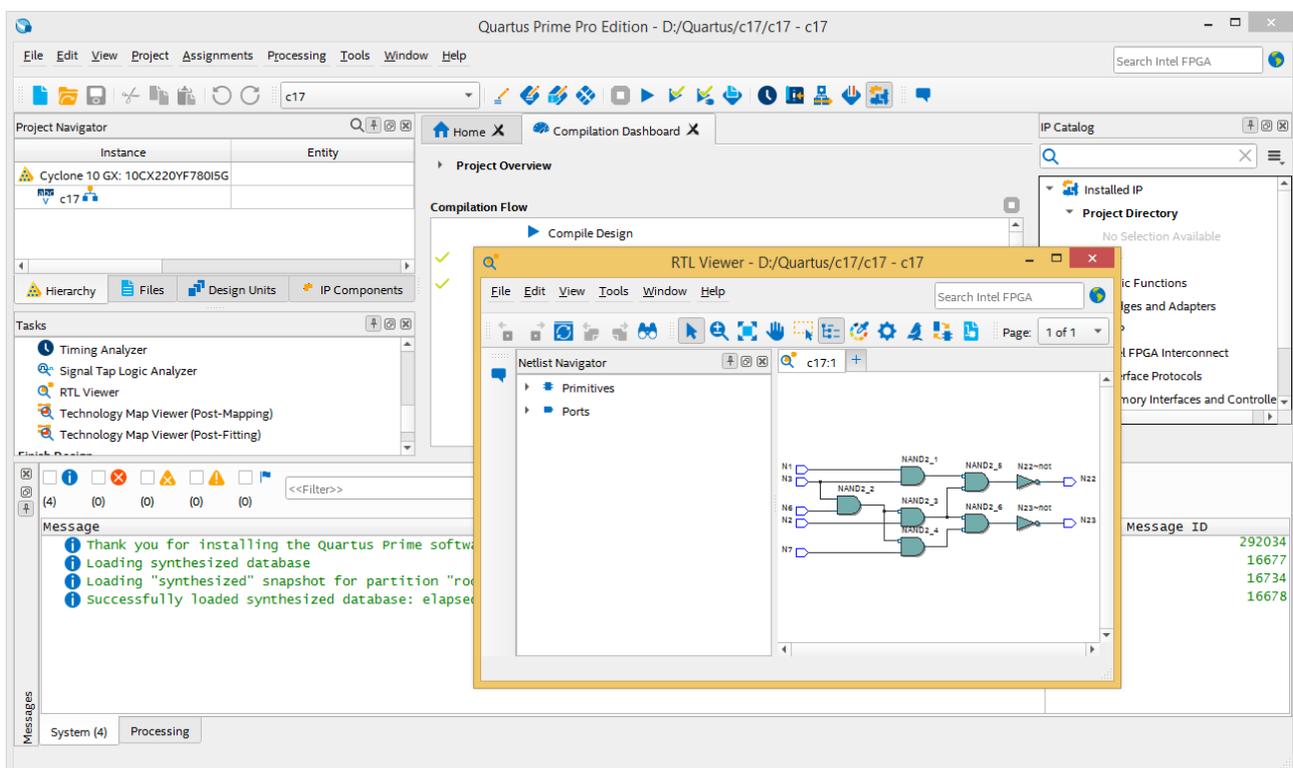


Рис. 1. Моделирование в EDA Quartus II тестовой схемы ISCAS c17  
 Fig. 1. Simulation in EDA Quartus II test circuit ISCAS c17

Это подтверждается многолетним опытом применения такого подхода при создании заказных схем на основе FPGA [6]. В качестве среды моделирования могут использоваться как проприетарные системы (proprietary system) EDA, например Quartus II или ModelSim, так и программы с открытым исходным кодом (open source), например пакет программ Iverilog. В первом случае это системы автоматизированного проектирования (САПР), ориентированные на разработку заказных микросхем на основе программируемых матриц. Производители этих САПР (Intel, Altera, Mentor Graphics, Xilinx и т.д.), предусматривают в них, как правило, три этапа проектирования: создание алгоритмической (электронной) модели будущей микросхемы; трассировку соединений элементов в FPGA; программирование логической матрицы (перенос программы трассировки в FPGA с помощью программатора, подключенного к EDA). Каждый этап проектирования сопровождается своим поведенческим контролем (simulation) и отладкой. При организации диагностических экспериментов вначале вполне достаточно ограничиться проведением функционального контроля, остальные два нужны для настройки технологического процесса изготовления готовых микросхем. Аналогичный этап содержится и в проектировании микросхем на основе Silicon IP [7] для подготовки документации и передачи ее кремниевой фабрике для производства микросхем. Упомянутые выше EDA имеют предпочтение перед последними САПР, так как имеют «облегченные» (студенческие) лицензии, которые предоставляют для диагностических экспериментов практически все функции EDA [8]. Кроме того, большинство форматов данных в современных EDA совместимы и дополняют друг друга. В случае необходимости недоступные функции можно реализовать open source программами, например входящими в комплект программ Iverilog. Вводить исходные данные в среде EDA можно с помощью традиционных эпюр, используя графический редактор САПР, или программируя их на языках Verilog, VHDL. Первый язык предпочтительней, так как Verilog более компактен и лучше адаптирован к EDA. После ввода элементов архитектуры и параметров ЦУ (для большинства ЦУ из ISCAS в свободном доступе уже имеется описание на языке Verilog, VHDL) программа компилируется и при положительном результате формируется файл, отображающий ЦУ на уровне регистровых передач – RTL (Register Transfer Level). Вызвав этот файл в среде EDA, можно графически проконтролировать схему ЦУ, она должна соответствовать требованиям ISCAS. Кроме контроля она еще является и подтверждением успешного завершения этого этапа.

## 2. Создание тестов на языке Verilog в среде EDA

Создание тестов (testbench) на языке Verilog в среде EDA, по мнению некоторых авторов [6], является «самой интеллектуальной работой», даже сравнимой с искусством. Эта работа трудно поддается автоматизации, ее трудоемкость сильно зависит от требований к тестовым наборам. Проще всего задать псевдослучайные или периодически повторяющиеся регулярные наборы (САПР их сама генерирует по задаваемым параметрам), сложнее сформировать функциональные контрольные тесты, так как не для всех тестовых схем ISCAS можно найти подробное описание их работы. Самыми сложными тестами являются диагностические, ориентированные на задаваемые модели неисправностей. Как правило, EDA testbench оформляют в виде второго файла (рис. 2), а первый – это модель ЦУ.

Например, в EDA Quartus II это VWF файлы, в Iverilog это V файлы (Verilog файлы). Современные EDA в testbench могут включать не только тестовые наборы, но и средства анализа и визуализации (программный вариант логического анализатора), управляемые графически или программно (Quartus II). Кроме того, EDA могут кроме функционального анализа (Functional Simulation) выполнять и временной анализ (Timing Simulation), а это уже охватывает не только логические неисправности (одиночные константные или кратные, но и динамические). Все это требует тестовых наборов, обеспечивающих выявление «симптомов» такого рода неисправностей. Создание сложных диагностических тестов и является одной из целей современной диагностики, предполагающей наряду с использованием традиционных и новые алгоритмы (эволюционные, нейроморфные, квантовые и т. д.). Именно поэтому и нужны современные автоматизированные средства оценки качества этих и вновь разрабатываемых алгоритмов построения эффективных тестов.

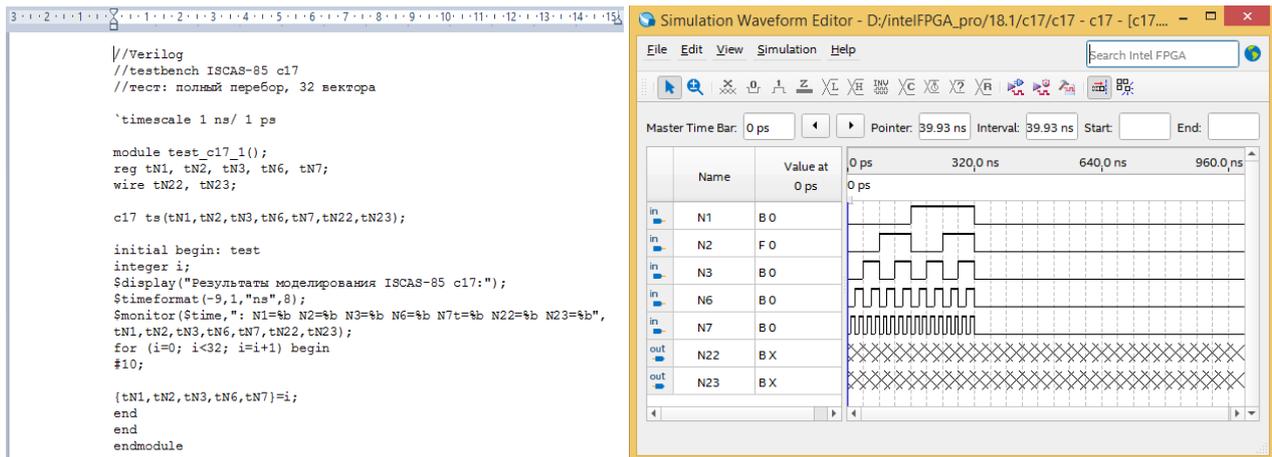


Рис. 2. Testbench схемы ISCAS c17  
 Fig. 2. Testbench circuit ISCAS c17

При использовании графических средств построения testbench и генерации тестовых наборов средствами EDA анализируется модель ЦУ. Визуализатор при этом может выводить шаблон логического анализа (горизонтальные оси и соответствующие им обнаруженные входы и выходы). Используя этот шаблон, можно графически задать (как бы «начертить») входные наборы, тем самым сформировать testbench. После создания модели ЦУ и testbench, как правило, оформленных в виде двух файлов, наступает этап компиляции и отладки. По существу, это итерационный процесс устранения ошибок в поведении модели ЦУ и тестах. Так как ошибки модели обычно устраняются на втором этапе, то на этом этапе в основном выявляются ошибки testbench, описание которых выводится сообщениями EDA. Реакция на эти сообщения и изменение testbench являются основной целью этапа отладки, затрагивающего как валидность самой модели, так и особенности анализируемого алгоритма построения теста. В результате такого итерационного процесса, когда сообщения об ошибках при компиляции не возникают, генерируются файлы внутреннего формата EDA (в некоторых EDA они доступны в каталоге модели) и выводятся результаты моделирования, если в меню визуализатора указывались соответствующие панели отображения.

Диагностические эксперименты заключаются в задании временных параметров работы симулятора, получении массива данных (data set), сгенерированного EDA, и отображении их в заданном формате. Так, в EDA Quartus II в панели Simulation Waveform Editor, где графически создается и testbench, можно задать временную шкалу моделирования и вызвать в меню процедуру функционального моделирования (Run Functional Simulation), выведя результаты на соответствующие оси. Причем в подкаталог Simulation рабочего каталога модели ЦУ будет выведен файл результата диагностического эксперимента, который можно в дальнейшем проанализировать и опубликовать (рис. 3).

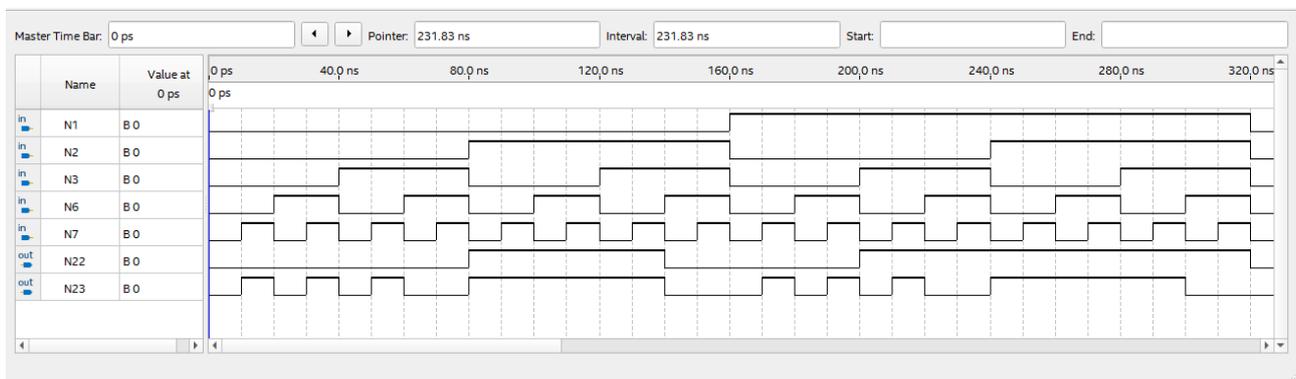


Рис. 3. Результаты диагностического эксперимента с тестовой схемой ISCAS c17  
 Fig. 3. The results of a diagnostic experiment with a test circuit ISCAS c17

В пакете программ Iverilog процесс моделирования запускается отдельной моделирующей программой, которая вызывает откомпилированный файл и выводит результат на экран или в указанный файл (рис. 4).

```

Результаты моделирования ISCAS-85 c17:
0: N1=x N2=x N3=x N6=x N7=t=x N22=x N23=x
10: N1=0 N2=0 N3=0 N6=0 N7=t=0 N22=0 N23=0
20: N1=0 N2=0 N3=0 N6=0 N7=t=1 N22=0 N23=1
30: N1=0 N2=0 N3=0 N6=1 N7=t=0 N22=0 N23=0
40: N1=0 N2=0 N3=0 N6=1 N7=t=1 N22=0 N23=1
50: N1=0 N2=0 N3=1 N6=0 N7=t=0 N22=0 N23=0
60: N1=0 N2=0 N3=1 N6=0 N7=t=1 N22=0 N23=1
70: N1=0 N2=0 N3=1 N6=1 N7=t=0 N22=0 N23=0
80: N1=0 N2=0 N3=1 N6=1 N7=t=1 N22=0 N23=1
90: N1=0 N2=1 N3=0 N6=0 N7=t=0 N22=1 N23=1
100: N1=0 N2=1 N3=0 N6=0 N7=t=1 N22=1 N23=1
110: N1=0 N2=1 N3=0 N6=1 N7=t=0 N22=1 N23=1
120: N1=0 N2=1 N3=0 N6=1 N7=t=1 N22=1 N23=1
130: N1=0 N2=1 N3=1 N6=0 N7=t=0 N22=1 N23=1
140: N1=0 N2=1 N3=1 N6=0 N7=t=1 N22=1 N23=1
150: N1=0 N2=1 N3=1 N6=1 N7=t=0 N22=0 N23=0
160: N1=0 N2=1 N3=1 N6=1 N7=t=1 N22=0 N23=0
170: N1=1 N2=0 N3=0 N6=0 N7=t=0 N22=0 N23=0
180: N1=1 N2=0 N3=0 N6=0 N7=t=1 N22=0 N23=1
190: N1=1 N2=0 N3=0 N6=1 N7=t=0 N22=0 N23=0
200: N1=1 N2=0 N3=0 N6=1 N7=t=1 N22=0 N23=1
210: N1=1 N2=0 N3=1 N6=0 N7=t=0 N22=1 N23=0
220: N1=1 N2=0 N3=1 N6=0 N7=t=1 N22=1 N23=1
230: N1=1 N2=0 N3=1 N6=1 N7=t=0 N22=1 N23=0
240: N1=1 N2=0 N3=1 N6=1 N7=t=1 N22=1 N23=0
250: N1=1 N2=1 N3=0 N6=0 N7=t=0 N22=1 N23=1
260: N1=1 N2=1 N3=0 N6=0 N7=t=1 N22=1 N23=1
270: N1=1 N2=1 N3=0 N6=1 N7=t=0 N22=1 N23=1
280: N1=1 N2=1 N3=0 N6=1 N7=t=1 N22=1 N23=1
290: N1=1 N2=1 N3=1 N6=0 N7=t=0 N22=1 N23=1
300: N1=1 N2=1 N3=1 N6=0 N7=t=1 N22=1 N23=1
310: N1=1 N2=1 N3=1 N6=1 N7=t=0 N22=1 N23=0
320: N1=1 N2=1 N3=1 N6=1 N7=t=1 N22=1 N23=0
    
```

Рис. 4. Результаты диагностического эксперимента с тестовой схемой ISCAS c17 в Iverilog  
 Fig. 4. The results of the diagnostic experiment with the test circuit ISCAS c17 in Iverilog

При необходимости можно создать и графический файл визуализации, который будет открываться в отдельной программе визуализатора. Причем в Iverilog все параметры моделирования указываются в testbench. Полученные данные (data set) и являются целью диагностического эксперимента, так как по ним можно судить о качестве тестов, созданных на основе анализируемого алгоритма. Сама среда EDA в такой диагностический эксперимент ничего от себя не вносит, она только позволяет создать адекватную модель ЦУ и тест на основе исследуемого алгоритма. При необходимости тест можно создать и другими средствами, а EDA использовать для его подачи. При этом упоминавшиеся выше EDA имеют возможность тестировать и физические объекты (ЦУ), подключаемые к ним через порты компьютера.

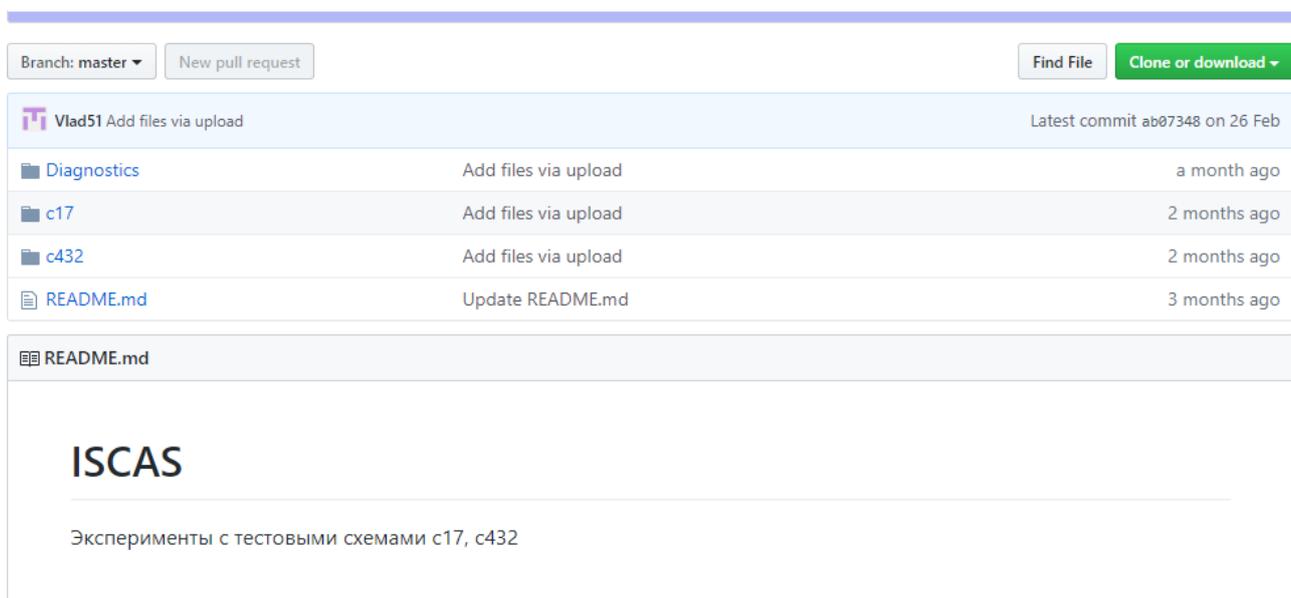


Рис. 5. Публикация результатов диагностических экспериментов в открытом доступе  
 Fig. 5. Publication of the results of diagnostic experiments in the public domain

Завершающим этапом в диагностическом эксперименте является анализ полученных результатов и публикация их в открытом доступе (рис. 5). Для этого можно использовать как публичные облачные сервисы, так и сервисы контроля версий типа GitHub [9]. Это позволит в дальнейшем неоднократно обращаться к ним в процессе работы над алгоритмом и представить конечный результат на суд экспертам.

### Заключение

Предложенная методика оценки качества контрольных и диагностических тестов позволяет объективно оценить качество алгоритмов построения тестов и автоматизировать диагностический эксперимент, проводимый на алгоритмических моделях тестовых схем ISCAS. Кроме того, результаты диагностического эксперимента, представленные как scientific data, имеют широко распространенные форматы, что облегчает их анализ и публикацию.

### ЛИТЕРАТУРА

1. ISCAS. 2019. URL: <http://www.pld.ttu.ee/~maksim/benchmarks/> (accessed: 15.10.2019).
2. Quartus II Introduction for Verilog Users. Altera Corporation, 2009. 30 p.
3. ModelSim : руководство пользователя. Model Technology Incorporated, 2002. 540 с.
4. Icarus Verilog. 2019. URL: <http://iverilog.icarus.com/> (accessed: 15.10.2019).
5. Эксперименты с тестовыми схемами. 2019. URL: <https://github.com/Vlad51/ISCAS> (дата обращения: 15.10.2019).
6. Соловьев В.В. Основы языка проектирования цифровой аппаратуры Verilog. М. : Горячая линия. Телеком, 2014. 206 с. ISBN 978-5-9912-0353-1.
7. Заказные блоки в микросхемах (Silicon IP): как это работает. 2019. URL: <https://habr.com/ru/post/414215/> (дата обращения: 15.10.2019).
8. Quartus II Handbook. Altera Corporation, 2015. Vol. 1: Design and Synthesis. 1800 p.
9. Chacon S., Straub B. Pro Git. Second ed. Apress, 2019. 518 p.

Поступила в редакцию 15 октября 2019 г.

Solovyev V.M., Speranskiy D.V. (2020) ASSESSMENT OF THE QUALITY OF DIAGNOSTIC TESTS ON MODELS OF DIGITAL DEVICES ISCAS. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 52. pp. 123–129

DOI: 10.17223/19988605/52/15

A methodology for evaluation of the quality of diagnostic tests on models of digital devices (DD) standardised by the IEEE International Symposium is considered. The need to develop such a technique is caused by the intensive process of introducing electronic products into all areas of human activity. These electronic products (digital devices) need to be designed, manufactured and operated, keeping them in a good working state. At each stage of the life cycle of the DD it is necessary to know their technical status (correct or faulty), which is ensured by monitoring and diagnostic, apply the tests and to analyze the DD reaction to it. Creating effective tests is one of the urgent tasks of modern diagnostics. The effectiveness of tests depends on the algorithms used to create them. Now there are many such algorithms, whose products (tests) require analysis and evaluation of quality. For an objective analysis and evaluation of algorithms experts developed test circuits of DD (ISCAS [1]), standardised by the IEEE International Symposium on Circuits and Systems.

Using these test circuits, it is possible in the computer-aided design of electronic devices (Electronic Design Automation, EDA) to create algorithmic models of the ISCAS of DD, which are almost the same as their actual physical models. Modern EDA, as a rule, allows you to create algorithmic models in the languages of the description of electronic circuits Verilog or VHDL. Previously developed models on VHDL can be translated to Verilog with subsequent compilation, which ensures their compatibility. At the same time, it becomes possible to effectively explore the various approaches and ideas underlying the test synthesis algorithms for complex DD. In addition, this approach allows us to evaluate the technical DD state to create a convenient apparatus for obtaining, storing and analyzing the results of diagnostic experiments. It is very important that these results can be presented in various convenient forms for the researcher (bit array, binary vector, graphic images, etc). All this makes EDA the basis for the development of a new promising methodology that solves a wide range of problems of technical diagnostics of complex DD. The capabilities of this technique were illustrated on standard ISCAS circuits «c17» and «c432». For ISCAS circuits the proposed methodology may include the following steps: selection of a test circuit from ISCAS; data input (schematic diagram or description of DD in Verilog, VHDL) and parameters in the simulation environment (for example, Quartus II [2], ModelSim [3], Icarus Verilog (Iverilog) [4], etc.); development on the base of analyzed algorithms tests for control and diagnostic in the simulation environment; compilation and debugging of the created

DD model and tests; diagnostic experiments with a well-functioning model DD and tests in the simulation environment, as well as visualization of the results of diagnostic experiment; analysis of the results and placing the results in open access to ensure the possibility of their verification and repetition [5].

Keywords: digital devices (DD); languages of DD description Verilog, VHDL; control and diagnostic of DD; programs of Electronic Design Automation (EDA).

*SOLOVYEV Vladimir Mikhailovich* (Candidate of Technical Sciences, Associate Professor, Head of the Volga Regional Center for New Information Technologies of Saratov State University, Saratov, Russian Federation).

E-mail: ign1122@mail.ru

*SPERANSKIY Dmitriy Vasilyevich* (Professor, Doctor of Technical Sciences, Department Railway Automatics, Telemechanics and Communication of Russian University of Transport (MIIT), Moscow, Russian Federation).

E-mail: speranskiy.dv@gmail.com

#### REFERENCES

1. ISCAS. (2019) [Online] Available from: <http://www.pld.ttu.ee/~maksim/benchmarks/>. (Accessed: 15th October 2019).
2. CSEWEB. (2009) *Quartus II Introduction for Verilog Users*. Altera Corporation.
3. Model Technology Incorporated. (2002) *ModelSim. User's Manual*. [s.l.; s.n.].
4. *Icarus Verilog*. (2019). [Online] Available from: <http://iverilog.icarus.com/>. (Accessed: 15th October 2019).
5. Github.com. (2019) *Eksperimenty s testovymi skhemami* [Experiments with test circuits]. [Online] Available from: <https://github.com/Vlad51/ISCAS>. (Accessed: 15th October 2019).
6. Solovyev, V.V. (2014) *Osnovy yazyka proektirovaniya tsifrovoy apparatury Verilog* [Fundamentals of the language of design of Verilog digital equipment]. Moscow: Goryachaya liniya. Telekom.
7. Habr.com. (2019) *Zakaznye bloki v mikroskhemakh (Silicon IP): kak eto rabotaet* [Custom blocks in chips (Silicon IP): how it works]. [Online] Available from: <https://habr.com/ru/post/414215/>. (Accessed: 15th October 2019).
8. CSEWEB. (2015) *Quartus II Handbook*. Vol. 1. Altera Corporation.
9. Chacon, S. & Straub, B. (2019) *Pro Git. Second Edition*. Apress.